Efficient and Reliable Network Tomography in Heterogeneous Networks Using BitTorrent Broadcasts and Clustering Algorithms

Kiril Dichev*, Fergal Reid*†, Alexey Lastovetsky*
*School of Computer Science and Informatics †Clique Research Cluster
University College Dublin, Belfield, Dublin 4,
Ireland

Kiril.Dichev@ucdconnect.ie
Fergal.Reid@gmail.com
Alexey.Lastovetsky@ucd.ie

***Abstract*—In the area of network performance and discovery, network tomography focuses on reconstructing network properties using only end-to-end measurements at the application layer. One challenging problem in network tomography is reconstructing available bandwidth along all links during multiple source / multiple destination transmissions. The traditional measurement procedures used for bandwidth tomography are extremely time consuming. We propose a novel solution to this problem. Our method counts the fragments exchanged during a BitTorrent broadcast. While this measurement has a high level of randomness, it can be obtained very efficiently, and aggregated into a reliable metric. This data is then analyzed with state-of-the-art algorithms, which correctly reconstruct logical clusters of nodes interconnected by high bandwidth, as well as bottlenecks between these logical clusters. Our experiments demonstrate that the proposed two-phase approach efficiently solves the presented problem for a number of settings on a complex grid infrastructure.**

***Index Terms*—Network tomography, BitTorrent, clustering, bandwidth, bottleneck link**

I. Introduction and Related Work

The properties of the underlying network play a central role in the performance of all distributed and parallel applications which rely on communication. When these properties are taken into account, communication can be optimized. In the Message Passing Library (MPI), every collective operation can profit through topology awareness, particularly in heterogeneous networks. A large body of research has been done in this direction for various protocols and networks, including but not limited to [1], [2], [3], [4], [5] . Existing work performs topology-aware collective operations using knowledge of a pre-defined partition clustering of the network, and finds that these topology-aware collectives substantially outperform topology-agnostic methods. In this work, our goal is to provide a versatile automated method to deduce a partitioning of the network into logical bandwidth clusters. We aim to provide a method that is efficient in its network measurement, and good at finding network bottlenecks under conditions of high load. This method would allow easy topology aware communication on large highly utilized heterogeneous networks, which are becoming an increasingly important domain for distributed computation.

There are two main ways of incorporating knowledge about the heterogeneity of the network into communication algorithms - by providing such knowledge manually or automatically. Past MPI implementations for Grid infrastructures and other wide-area networks [1], [2], [3] have used each of these ways of incorporating such knowledge. Fully automatic approaches can be subdivided into 'intra-node' and 'inter-node' approaches. For inter-node approaches, network discovery typically involves some form of communication model. There are many examples of this approach in high-performance computing and early work includes [6], [7]. For intra-node automatic approaches, which are gaining popularity today due to many-core nodes, recent work includes [8].

However, the existing automated approaches in high-performance computing do not generally capture all relevant network properties. Examples of properties that existing approaches are poor at capturing include bottleneck links; as while often not visible in isolated point-to-point communication, these bottlenecks appear under conditions of particularly intense collective communication. We look for possible solutions to these peak-bandwidth measurement issues in distributed computing rather than high-performance computing since heterogeneous and hierarchical networks are traditionally used in this domain. The challenges presented by such networks gave rise to an interesting sub-field of network discovery in the late 90s – 'network tomography'. Castro [9] provides a detailed overview. The general goal of network tomography is to reconstruct the logical topology of the network in two phases, as depicted in Fig. 1. The first phase involves only end-to-end measurements of the network. Based on how these measurements are performed - whether actively or passively - we can talk of passive network tomography (e.g. [10], [11]) or active network tomography (e.g. [12], [13]). After measurement data is collected, the second phase

SC12, November 10-16, 2012, Salt Lake
978-1-4673-0806-9/12/$31.00 ©2012 IEEE

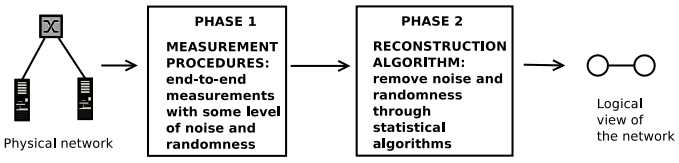
PHASE 1
MEASUREMENT PROCEDURES: end-to-end measurements with some level of noise and randomness
PHASE 2
RECONSTRUCTION ALGORITHM: remove noise and randomness through statistical algorithms
Physical network
Logical view of the network

Fig. 1. Overview of network tomography.

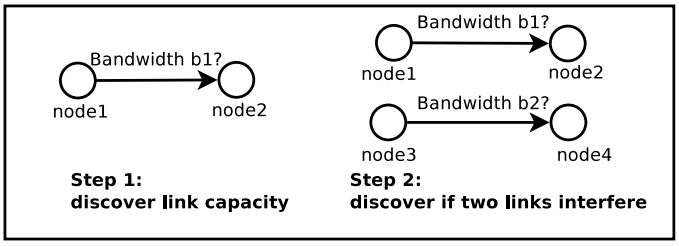

Fig. 2. Traditional measurement of bandwidth in tomography.

of the process always involves the use of statistical methods to reconstruct the logical view of the network. While many metrics can be used, a number of metrics are particularly relevant from the user perspective, including loss rate, delay and bandwidth.

The main contribution of this work is in the area of active network tomography with a bandwidth-related metric. Our focus is multiple source / multiple destination communication. An example of such communication would be heavy bulk transfer between all network peers (e.g. in all-to-all communication). The closest related work to ours are [14] and [15]. Both works reconstruct the logical topology of the network using bandwidth as a metric. While [15] infers a qualitative view of the network, [14] infers a more quantitative view, including labeling of actual achievable bandwidth.

The most time-consuming phase in existing approaches to active network tomography on bandwidth is the measurement phase. The measurement procedures used to find the available bandwidth and/or bottlenecks between communicating peers are generally similar across all tomography methods. We show the two essential steps that are generally involved in Fig. 2.

Conceptually, in the first step, an intense communication is established between a pair of nodes until the link capacity between them is reached. Then, a new pair of intensely communicating nodes is introduced, and the bandwidth of the first link is reexamined. If no change in bandwidth is observed, then the links are probably independent – more pairs communicating in parallel could unveil a bottleneck at a later point. But if the bandwidth of the node pairs under examination decreases, then it is clear that they share the same physical (and logical) link. Following this procedure, experiments are performed until the entire network is reconstructed. While intuitive, this approach is very expensive. The measurement procedures have polynomial complexity, even after some optimizations using heuristics or parallelism. Indeed, this complexity makes it infeasible to perform bandwidth-related network tomography on large-scale computer networks. Both [14] and [15] concede this issue and resort to running simulations using SimGrid [16]. [15] only attempts real experiments on a small scale. Even simplified measurement procedures require approximately one hour to run with only 20 nodes.

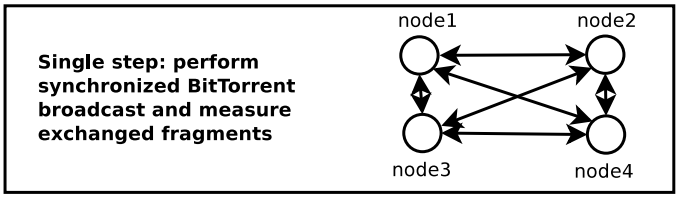

Fig. 3. Alternative bandwidth-related measurement as proposed in this work.

In our work, we introduce a novel approach to multiple source / multiple destination network tomography, which differs from traditional approaches described in Fig. 1 in both the measurement procedure and the analysis method as follows:

- Phase 1: We use the BitTorrent [17] protocol as the tool for our measurements. The measurement metric is the number of exchanged fragments between peers during a synchronized BitTorrent broadcast (see Fig. 3).
- Phase 2: In the second phase, we use a popular implementation of a modularity-based network clustering algorithm [18] [19] to perform analysis of the measured data. We describe this process, and choice of clustering algorithm, as well as its operation, in detail in Section III-B.

We are not aware of previous work in this area using the type of measurement procedure we employ, nor are we aware of any coupling of this approach with a clustering algorithm in the manner which we present.

The intuition behind the metric is that when using a number of parallel connections, more data will be naturally transferred through the links with higher bandwidth. This idea has been demonstrated even before the advent of protocols like BitTorrent. For example, [20] demonstrated that a client using a number of parallel TCP connections to servers with different upload rates will download a file with a rate approaching the fastest upload rate. Indeed, the BitTorrent protocol uses a number of parallel connections to exploit this network feature. The proposed measurement departs from existing approaches in the field not only in the metric, but also in its efficiency - a single synchronized BitTorrent broadcast can often capture the behavior of a very large number of links. Indeed, every broadcast of a large message to many peers has a total communication time linear in the message size. The number of peers does not reduce the download rate due to the pipelining and scalability of the BitTorrent algorithm. For detailed performance analysis of BitTorrent see for example [21], [22].

Surprisingly, we find only one other work [23] which investigates BitTorrent tomography. However, this work states that BitTorrent traffic is immeasurable on a large scale. The authors do not list any technical issues with this approach, but argue that it is unlikely for an instrumented BitTorrent client to be used by a large user community. They replace BitTorrent profiling with an alternative algorithm, which they

use in conjunction with a simulation program. There is also work which proceeds in the opposite direction - optimizing the performance of the BitTorrent protocol, given knowledge of the network topology (e.g. [24]).

In order to validate our tomography method we perform a set of experiments on real networks using a grid infrastructure. Although we focus on bulk transfers of large data, our measurement approach is efficient, because it does not perform exhaustive measurements. The measurement procedure, which consists of several iterations of BitTorrent broadcasts, captures the flow of large data volumes across the entire network in each iteration. With just a few iterations, we find that sufficient information can be collected to infer important network properties. This information can be used in many data-intensive communication operations. For example, if we want to efficiently schedule an all-to-all operation, we do not need to label the achievable bandwidth along all fast and slow links in the network. Instead, the only requirement is a logical clustering of nodes according to their bandwidth. Thus, if there is a bottleneck link between nodes, a correct clustering algorithm should place them in different logical clusters.

We find that our method efficiently and reliably clusters nodes with regard to their bandwidth when all nodes are involved in collective communication. We use the algorithm successfully both for experiments separating compute clusters within a site, and between sites, on the Grid'5000 infrastructure.

The paper is structured as follows: Section II presents our metric and measurement procedures. Section III presents the clustering algorithm we use. In Section IV we present results from a range of experiments. We conclude the paper in Section V.

II. BitTorrent Broadcasts and "Exchanged Fragments Between Peers" As a Metric

A. Definition and Use

Throughout this paper, we refer to a BitTorrent broadcast as a *fully synchronized instrumented execution of BitTorrent clients until all clients have downloaded a file*. The used metric is derived from these broadcasts and is bandwidth-related. We observe the communication network as a directed graph G=(V,E). A file of size M is distributed as $\frac{M}{16KB}$ fragments of 16KB to all nodes v $\in$ V using BitTorrent broadcasts. If $v_1 \rightarrow_i v_2$ denotes the number of fragments sent directly from v_1 to v_2 within broadcast operation *i*, then we define the metric *w* per edge *e* for one run as

$$w(e) = v_1 \rightarrow_1 v_2 + v_2 \rightarrow_1 v_1 \tag{1}$$

with $e = (v_1, v_2)$. Since performing more iterations significantly increases the accuracy of the metric, for *n* iterations we simply state

$$w(e) = \frac{\sum_{i=1}^{n} v_1 \rightarrow_i v_2 + v_2 \rightarrow_i v_1}{n} \tag{2}$$

with $e = (v_1, v_2)$.

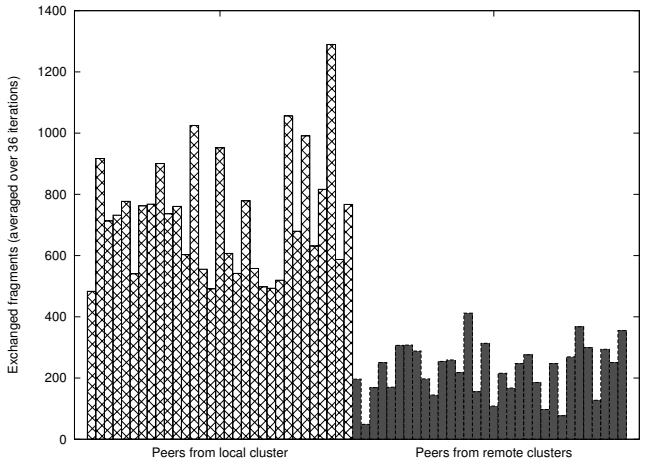

Fig. 4. *w(e)* as measured for all edges *e* to a randomly fixed node (36 iterations). On the left are edges to local cluster nodes, on the right are edges to remote nodes.

In this work, the size of a file used in a single BitTorrent broadcast is chosen to be 239 MB. This choice is completely arbitrary and is driven by practical observations that a single broadcast then takes around 20 seconds for different numbers of nodes (see Section II-B for details). We find this to be a reasonable amount of time for a single broadcast, that often provides good bandwidth information for many of the available links. Profiling the precise number of fragments exchanged gives the following information: in each BitTorrent broadcast, exactly 15259 fragments of 16384 bytes each are received by all participating nodes, following a dynamic pattern each time.

We have instrumented the original Python version of the BitTorrent client written by Bram Cohen and available in most Linux distributions. We introduce efficient profiling of the arriving data as follows: At the reception of each data fragment, a counter is incremented associated with the sending peer using a hash table of counters. At the end of a run, all peers have a record of the source peers and the number of fragments they received from each peer.

As an example, in a broadcast operation involving 64 nodes on one site, we display measurements for a randomly chosen node in Fig. 4. The bars represent the metric as defined above for 36 iterations for all edges which include the fixed node. Since the results involve many iterations, the chosen node exchanges fragments with all 63 peers. For clarity, we have grouped on the left side the metric values for the 31 peers in the local cluster, and grouped the values for the 32 remote nodes on the right side.

We will discuss the main characteristics of this measurement, and explain how it differs to classic bandwidth measurements in the following sections.

B. Efficiency of the Metric

The main strength of our method is that it takes only a single broadcast of a large message per run to collect data

on a large subset of all possible peer-to-peer connections. In our setup, the observed complexity of BitTorrent broadcasts is *O(M)* – linear in the message size *M*. We verified experimentally that as we alter the number of nodes, the BitTorrent broadcast requires nearly constant time. According to practices from high-performance computing, our reference time for the completion of a BitTorrent broadcast is the maximum download completion time of all the BitTorrent clients, which we start synchronously. For 32, 64 and 128 nodes, the broadcast of the 239 MB large message takes about 20 seconds on the Grid'5000 infrastructure, even when the nodes are spread across 4 sites. Related work [22] also suggests that a high download rate can be sustained for very large peer numbers; the number of participating peers in such experiments typically does not alter the estimated time of *O(M)* for all peers to download the file. Other work [25] also demonstrates that the BitTorrent protocol is competitive with client/server architectures in its peak download rate.

We now give a short overview of the complexity of the network tomography algorithms presented in the previous section. Each step in these algorithms – shown in Fig. 2 – is very time consuming. First, every link has to be saturated until the maximum bandwidth on that link is reached. This is a costly operation which incurs heavy network overhead. The second challenge consists of probing the link bandwidth in parallel for multiple links. This process is repeated until all nodes have been sufficiently tested. For example, [14] performs such tests only with at most triplets of nodes. It is stated that triplets are sufficient as long as the single-link experiments can reach the maximum capacity. Even with this assumption, all possible triplets need to be tested in the worst case. This step is performed since it is assumed that there is no a priori knowledge of the topology of the network. The observed complexity of the algorithm in this case is $O(N^3)$, in N the number of nodes.

The algorithm proposed by [15], on the other hand, tests pairs incrementally, fully in parallel and without limiting the maximum number of tested links at a time. In specific cases, where no interference of links is observed, the complexity is estimated at $O(N^2)$. The only empirical experiments performed are for networks of 20 nodes, and these take about one hour to complete.

C. Level of Randomness With Single Runs Using the Metric

If we examine the volume of exchanged data shown in Fig. 4 over a number of iterations, we notice that a total of 22533 fragments are exchanged with local cluster nodes, and 6337 fragments are exchanged with remote nodes. This is a clear indication that with BitTorrent broadcasts, data flows with a preference for high bandwidth links. Furthermore, we observe this phenomenon quite reliably in our experimental data.

We have previously defined a single run of our metric as transmitting a single file which takes approximately 20 seconds. As the operation of the BitTorrent protocol is stochastic, and the data transferred across each link varies from run to run, it is important to attempt to characterize the accuracy of the metric we have defined for a single run. Thus, we now observe how the metric fluctuates using one Grid'5000 site (Bordeaux). We focus on an edge between 2 nodes randomly chosen from within a cluster. Each run measures the metric $w(e)$ independently (no aggregation is used). Fig. 5 shows the distribution of $w(e)$ along the fixed edge over 36 runs. In 13 of the 36 runs, the two peers do not exchange any data with each other. In the other 23 runs, the exchanged data varies between 3 and 6304 fragments. This distribution shows that the variance is very high. For comparison, when running the well known NetPIPE tool [26] to establish the maximum achievable bandwidth along the link between two peer nodes on the same compute cluster used above, the variance is very low and the distribution is dense around 890 Mbps.

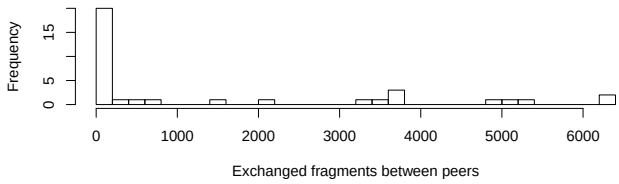

Fig. 5. Distribution of measured metric *w(e)* for a fixed edge *e* and 36 independent iterations.

Fig. 5 suggests that while inexpensive to compute, the metric is very variable for single runs. With this level of measurement noise and randomness, a good analysis technique will be needed to extract meaningful data from these measurements.Yet one important consideration is that our analysis method does not consider each link's bandwidth in isolation, as in previous approaches, and this eases to some extent the requirements for the measurement step.

Before discussing how we will aggregate and analyze data from this metric to form a reliable view of the network, we will briefly discuss the BitTorrent properties which are responsible for the variance and high degree of randomness between single runs of the metric:

- Initially, BitTorrent clients *randomly* choose their initial peers (adjustments in the peer selection are part of the protocol for longer runs).
- BitTorrent internally limits the number of parallel uploads to 4, and this indirectly limits the number of parallel downloads.
- Another protocol feature is that the number of total peers is limited to 35. This means that for larger numbers of nodes and a single broadcast, measurements using this protocol will not provide a complete graph - only a subset of possible connections will be measured. One solution to this problem is to aggregate the measurements over a number of BitTorrent broadcasts, as we shall see.
- Using a BitTorrent broadcast operation means that nodes which are better connected to the 'root node' are more likely to receive more fragments from the root. This

is simply due to the asymmetric way data flows in a broadcast operation as compared to, for example, an all-to-all transmission. However, in our experiments this was never an issue during the reconstruction and analysis of our networks. If this affects results in some cases, a simple solution is using different root nodes over a number of runs.

These are characteristics of the protocol, which, while important for transmission efficiency and reliability, increase the variance of our measurements, and could make network reconstruction hard. However, it will become evident in the following sections that despite the high degree of inherent randomness, this metric can easily be made reliable through simple iteration, especially when the measurements are analyzed using a clustering algorithm which operates on the observed network as a whole.

D. Improving the Accuracy Through Iteration of BitTorrent Broadcasts

While a single broadcast measurement has a high level of noise and randomness, aggregating data over a number of iterations resolves these issues. A positive property of the used approach is that each new iteration potentially improves the accuracy of the metric on a global scale – i.e. for all edges. The previous approaches presented can not address this, and are restricted to local experiments on a small subset of nodes. In order to quantify the number of iterations needed to improve our accuracy, the key questions are:

- How close is the single run data to an "ideal" representation of the peer-to-peer bandwidth when performing bulk data transfers?
- How fast does the aggregated data over a number of runs converge to the "ideal" representation?

We address these questions in an end-to-end manner, by quantitatively evaluating the performance of the entire system which uses these measurements. We make experimental observations as follows: after obtaining measurements at each run, we use a clustering metric – presented in detail in Section III-D – to assess the quality of measurements against a ground truth.

These questions could also be addressed by an analytical approach, but there are a number of challenges with this: the implementation of BitTorrent is not trivial to analyze, and porting it to a simulation environment is a complex engineering challenge.

III. Clustering Method

The previous section has shown that the used measurement approach can be very efficient, but that there is a significant level of randomness in the data gathering process. This would seem to pose a significant challenge for the clustering algorithm.

A second challenge is that we want to provide as little *a priori* information to our tomography method as possible, in order to increase the range of application scenarios. In particular, we do not want to specify the number of logical clusters into which to partition the observed network. We want to be able to deploy this method in real world application domains where very little information on the underlying topology is provided.

A. Cluster Visualization

In order to investigate the potential for an algorithmic clustering method to deduce the ground truth clusters from the measured network, we first visualise the measured network data using a network layout algorithm. On each layout visualisation, we use nodes of different shapes to represent different ground truth clusters. (The exact details of how the ground truth is produced from the physical network topology will be discussed in Section IV-A). As is shown visually in Figures 9, 10, 11, 12 and 13, the application of a layout algorithm to the measured networks allows us to visually observe groups of nodes the layout places close to each other. These groups clearly correspond to the 'ground truth' logical clusters of the underlying computer network.

For these visualisations, we layout the networks using the implementation of the Kamada-Kawai spring weighted graph layout algorithm [27] from the 'Graphviz' software package [28], in which we make the length of edges between nodes inversely proportional to the measured edge weight (which in turn corresponds to the presented metric *w(e)* – the exchanged fragments between two nodes). While we use all of the measured edges in our layout algorithms, for clarity of presentation in these diagrams we only render the edges which are in the top 50% of network edges by weight. It can be clearly seen that the ground truth clusters correspond to visually identifiable groups of nodes formed by the spring weighted layout. The fact that clear groupings are present in the layout visualisations is strong evidence that the BitTorrent measurement process is working correctly, and that the randomness in the data gathering process is not a problem when detecting groups. That the groupings correspond well to the ground truth clusters indicates that algorithmic clustering approaches will be successful on this problem.

B. Modularity-Based Clustering

That the layout visualizations show a relationship between the ground truth partitions and the groups of the nodes in the measured network indicates the measurement technique is working. However, visual inspection of observed clusters is not a robust means of evaluating performance. In addition, we want to be able to automate our tomography technique and to deploy it on networks too large to visualise. Thus we need an algorithmic clustering technique that finds clusters of nodes like those groups apparent on the layout visualisations.

To do this, we apply a technique from modern network analysis. We use the modularity function of Newman and Girvan [19] to identify sets of nodes which are more densely interconnected than the general level of interconnection in the network.

The modularity method is defined by the following objective [19]:

$$Q = \sum_i \left(e_{ii} - a_i^2\right) = Tr(e) - \left\|e^2\right\| \quad (3)$$

which compares, for a given clustering, the proportion of network edges that are intra-cluster e_{ii}, for each cluster i, against the proportion that would be intra-cluster in a randomized model of the same network. As described by Newman and Girvan: "This quantity measures the fraction of the edges in the network that connect vertices of the same type (i.e., within community edges) minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices."

We use a weighted version of this same objective, which will have a high value for clusters of nodes that have a high internal weight. This objective has been applied in a wide range of domains, including finding communities of users in social networks, finding highly connected communication groups in telecoms networks, and many other related application problems. As our objective is to find a partition of the network into dense non-overlapping clusters, and in particular as we do not wish to specify beforehand the number of logical clusters to find, this objective function is appropriate. In addition, our empirical results show it is effective at recovering the ground truth clusters as part of our tomography approach, as we discuss in Section IV.

C. Fast Louvain Method

Many different algorithms have been developed to optimize the modularity objective function. These algorithms improve on the original methods provided and are designed to work in practical settings and on large scale networks. One of the most successful and widely used methods is that of Blondel et al. [18], known as the Louvain method. This algorithm was originally developed and applied to large mobile telecommunications networks, in order to uncover clusters of frequently communicating users, and social communities; the authors found that they could uncover many levels of hierarchical organizational structure within the communications network.

While no meaningful close form complexity of this heuristic implementation is currently available, its fast runtime in practice and ability to scale to large datasets, such as telecoms networks with millions of nodes, make this modularity optimization algorithm suitable for our purposes.

This algorithm produces a dendrogram of hierarchical clusters by default. We do not use this dendrogram in this work; instead, we take the cut of the dendrogram at the point that yields the highest modularity value of the resulting partitions. This results in only a single level of partitioning. This is suitable for our purposes, as the ground truths which we use are partitions of the network; while the underlying networks themselves may be hierarchical, for simplicity and to fit our application domain we limit ourselves to non-hierarchical ground truths. However, there is potential for extending this approach in future, to uncover structure of a more hierarchical nature.

There are additional reasons to use the modularity maximisation method. The work of Noack [29] has shown an equivalence between modularity-based network partitioning approaches and particular types of force directed network layout algorithms. This does not include the Kamada-Kawai algorithm we use, but does provide motivation for trying the modularity-based methods in domains where graph layout algorithms successfully lay out nodes corresponding to their ground truth clusters.

We also attempted to perform these experiments with another modern clustering algorithm – Infomap [30] – which is based on compressing random walks through the network, and finds communities which correspond to the areas of a network that a random walk would get 'stuck' in. However, we find Infomap does not perform as well as modularity-based clustering on this particular problem.

Modularity maximisation is not without its problems: Good et al. [31] performed analysis of the modularity objective function in a variety of practical contexts, and concluded that the optimization surface is often bumpy, and often lacks a clear global maximum in empirical settings. However, we find that this widely used community finding algorithm produces results that work well in this particular application domain. Further, we find that repeated iterations of the optimization algorithm find results that are consistent with those presented in this paper; on the experimental networks we have examined, the algorithm seems to consistently converge to results that are in high agreement with our ground truth.

D. Comparing Network Clusterings

In order to quantitatively evaluate the performance of our method a numerical measure of clustering accuracy is necessary. Various methods for comparing set assignments exist. In the domain of network community finding, a frequently used measure of comparison between a ground truth clustering, and an algorithmically provided clustering, is the *Normalized Mutual Information* between the two. For convenience, and to enable the future extension of our work to situations where the ground truth overlaps, we use the overlapping NMI implementation of [32]. This method is capable of calculating the NMI between sets of communities which overlap, as well as sets of network partitions. This widely used measure enables us to compare our clustering against the ground truth. It ranges from 0 to 1, where 1 denotes perfect agreement of the found clustering with the ground truth. We note that there are several improvements on this NMI method. We have also investigated the results of some of these, and observed consistent results. As such we report scores only for the popular NMI method of [32].

Figure 6 shows the NMI values of our experimental work – to be described in detail in Section IV. Overall, it demonstrates that for various settings data aggregation converges to the ideal representation in a few iterations.

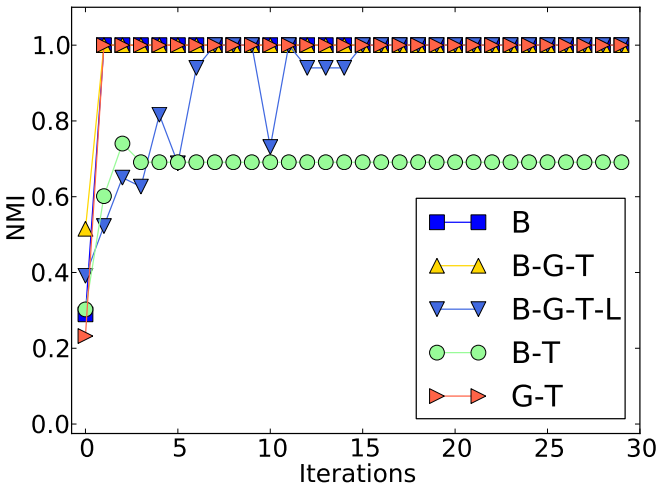

Fig. 6. Comparison of the clustering found using our tomography method, against the ground truth clustering provided. The results are shown in terms of Normalized Mutual Information [32]. We observe that, in general, the NMI improves as the number of measurement iterations performed increases, converging on a stable value. The convergence occurs quickly on the simpler topologies. The NMI frequently converges to 1 – perfect agreement with the ground truth. In the case where NMI does not converge to 1, visualized in Figure 10, we can see that a hierarchical ground truth, and clustering approach, may improve this. Details of the topologies between the four sites used – **B**ordeaux, **G**renoble, **T**oulouse and **L**yon – are provided in Section IV-A.

IV. Experimental Results

A. Introduction to Experimental Setup

The purpose of a network tomography method is to correctly uncover the properties of interest. In this particular work, the described algorithm performs well if the reconstruction is correct with regard to the dynamic bandwidth properties of the network. In practice, the relationship between these dynamic properties and the physical structure of the network topology is often complex. However, in order to evaluate our method, we use the physical structure of the network topology, including information about how network hardware connects compute clusters within physical sites, and information on the speed of the inter-site links, to form a ground truth dataset.

We perform our experiments on the Grid'5000 infrastructure in France. Nine sites are interconnected using the Renater network (Fig. 7) providing high bandwidth optical fiber. Within each site, there are differing technologies, hierarchies and clusters. For this work, only the Ethernet network within sites as well as the Renater network between sites is used.

The a priori knowledge of the network, which is independent of the network tomography algorithm, is very important in this work. This knowledge provides our ground truth which we use to evaluate the found clustering. We have ground truth information about multiple aspects of the system:

- The communication between sites has similar properties - it uses the Renater infrastructure. While it provides very high inter-site bandwidth, it is reasonable to assume it will not outperform local Ethernet communication. Experiments using NetPIPE confirm this assumption - for example, the maximum bandwidth achieved between nodes on Bordeaux and Toulouse is around 787 Mbps - compared to 890 Mbps achieved within Ethernet clusters.
- Within a Grid'5000 site, intra-site communication is complex. Physical hardware information is typically provided by online documentation available at [33]. However, transient network anomalies can arise when observing the network behavior (e.g. bandwidth bottlenecks, availability of multiple Ethernet interfaces, hardware changes), and so the authoritative ground truth clustering is generally best provided by the site administrator.

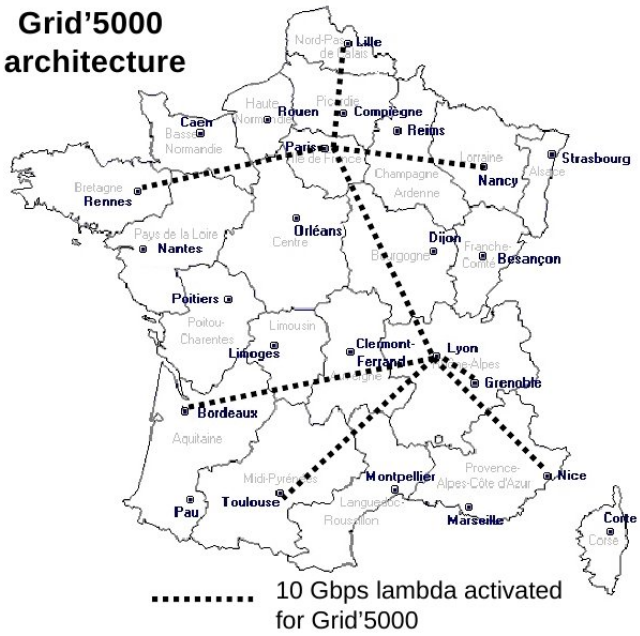

Fig. 7. The Renater infrastructure as presented in [33]

When we use a setup spanning multiple sites, we assume the clustering should subdivide the network into separate logical clusters, each cluster corresponding to a single site. If we evaluate our method on a single site – which we do for the Bordeaux network – we generate our ground truth using the available information about the structure of the physical topology in that site. We discuss these specifics in each of our experiments in turn.

B. One Site Experiments

In Fig. 8, we display a partial view of the network topology in Bordeaux, excluding the connections to the external network and the Myrinet / Infiniband network. One important realization is that even when provided with an explicit diagram of the network, it still is not obvious where the bottlenecks and the strong links are in terms of achievable bandwidth. The site administrator clarified that the significant bottleneck is the link between the Dell and Cisco switches, which only provides a single 1 Gigabit Ethernet connection. Note that the

link is only a bottleneck for the multiple source / multiple destination scenario we are addressing in this work.

1) 2x2 nodes: We start with a small experiment within the Bordeaux site with 2 nodes on the Bordeplage compute cluster and 2 nodes on the Borderline compute cluster. We ran 30 iterations and aggregated the measured data. The measurements provide very similar metrics for all links. For such a small setting, the link connecting Bordeplage and Borderline is not a bottleneck. In agreement with this observation, the used method identified a single logical cluster containing all four nodes.

2) 32x32 nodes: In another experiment we use 64 nodes - 32 nodes on Bordeplage, 5 nodes on Borderline and 27 nodes on Bordereau. We performed 36 BitTorrent iterations. Fig. 9 shows the visualisation of the results, which produce a perfect match to the real topology as displayed in Fig. 8. The two clusters Bordereau and Borderline (in circles) are merged together since they do not have a bottleneck link between them. However, the Bordeplage cluster (in diamonds) forms a different logical cluster, since it communicates to Borderline and Bordereau on a bottleneck 1 Gigabit link.

We also present the NMI between the specified ground truth clustering and the clustering produced by our tomography technique. Fig. 6 shows that after only 2 BitTorrent measurement iterations, the clustering is completely in accordance with the ground truth, and remains so during all additional iterations.

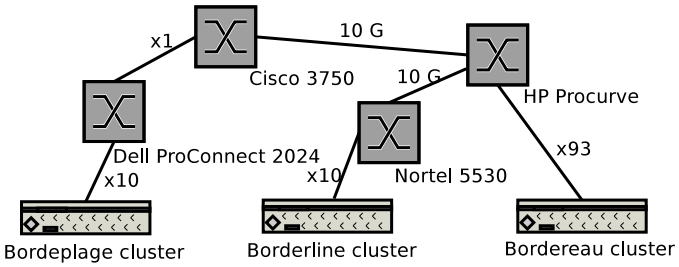

Fig. 8. Ethernet network at the 3 used clusters in the single site Bordeaux

C. Two Site Experiments

In the next step we extend the experiments to include nodes from two sites – Bordeaux and Toulouse. We still use 64 nodes in total – 32 nodes per site. We described the available bandwidth within Bordeaux in Fig. 8. For inter-site connections between sites on Grid'5000, the optic fiber Renater network is used. This connection provides very good bandwidth (10 Gbps) for inter-site communication, but overall the observed inter-site bandwidth is slightly lower than the intra-site bandwidth as described in Section IV-A. With the aggregated metric data, the clustering algorithm identifies two logical clusters, one corresponding to each of the two different sites.

Figure 6 shows that after 4 iterations, the clustering converges to a steady state. However, we note that the NMI with the ground truth, while high, is imperfect – approximately 0.7.

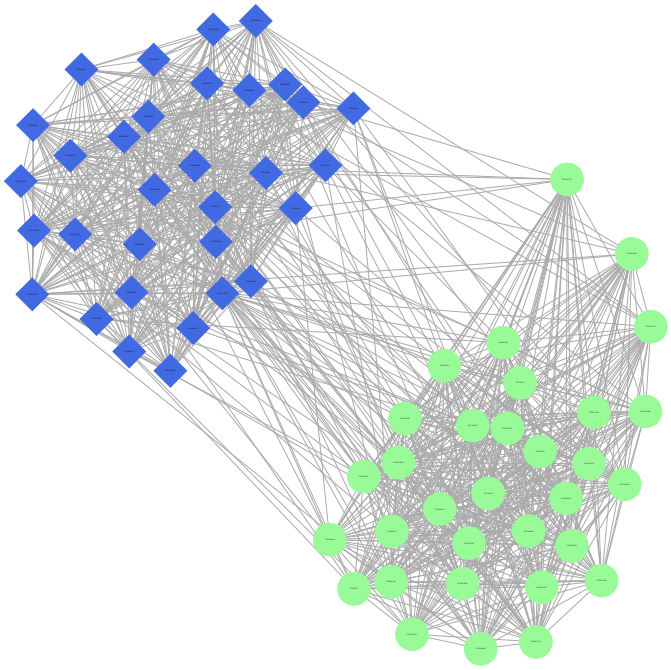

Fig. 9. Applying Kamada-Kawai layout (using the Graphviz' 'Neato' tool) to dataset 'B', the Bordeaux site. The configuration used has 64 nodes, divided between 3 physical compute clusters. These 3 physical compute clusters give rise to only 2 logical network clusters, as there is a fast link between the 'Bordereau' and 'Borderline' physical clusters. The shape and color of each node rendered reflects the labelling of the ground truth cluster it is in. We render only the edges in the top half of all edges, by weight. While the graph is too dense to visually make out any structure due to edge weight, it is clear that the layout algorithm is grouping nodes corresponding to their ground truth. This provides grounds for expecting a graph clustering algorithm to find these clusters.

On investigation, we observed that this is because we have provided a ground truth within which there are 3 different partitions: for the ground truth, the network was partitioned into the Bordeaux and Toulouse sites, and then the Bordeaux site was partitioned into two separate logical clusters (as discussed in the previous section), giving a total of three separate clusters. The best way to represent this physical setup is probably with a hierarchical representation of the clustering; however, in this work, to allow simple use of our results, we have chosen to focus on finding clusters which partition the network into a single level of clustering.

Fig. 10 shows that the Kamada-Kawai layout appears to correctly group Bordeaux and Toulouse, but also seems to layout two groups within the Bordeaux cluster. That the visualization makes visible the two separate sites within the Bordeaux cluster suggests that a future hierarchical version of our clustering step should be able to identify individual clusters within sites, at many levels, and makes clear the reason for the lower NMI in this case.

In another two site experiment, we used the sites Grenoble and Toulouse, again using 64 nodes and 30 runs. Unlike Bordeaux, Grenoble and Toulouse both have a very flat Ethernet network hierarchy within them.As such, neither Grenoble nor Toulouse are subdivided in our ground truth. The aggregated

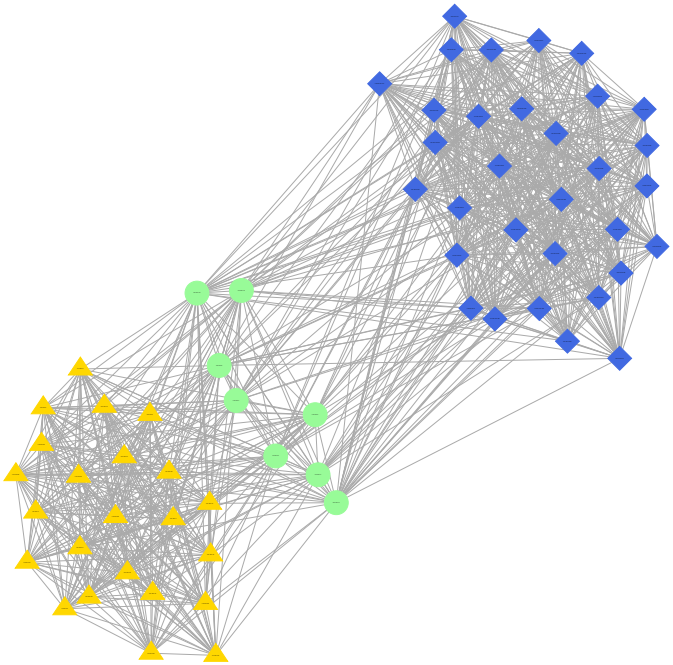

Fig. 10. Applying Kamada-Kawai layout to dataset 'BT', a set of nodes in Bordeaux and Toulouse, using the same rendering options as for Figure 9. Toulouse is represented here by diamonds; the ground truth clusters represented by circles and triangles both belong to Bordeaux. Our non-hierarchical clustering method does not recover this ground truth; it finds only two clusters, one for Toulouse and one for Bordeaux. The third ground truth cluster is distinct in the visualization, however, showing that the BitTorrent measurements do reflect it.

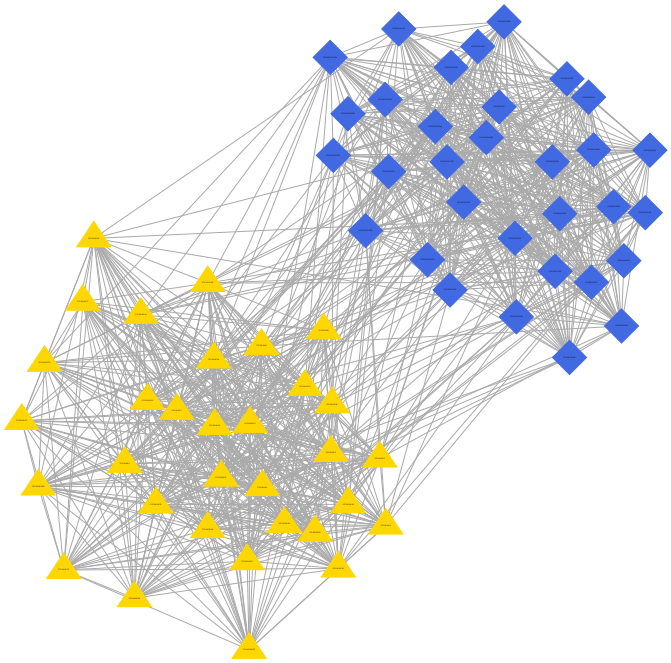

Fig. 11. Applying Kamada-Kawai layout to dataset 'GT', a set of nodes in Grenoble and Toulouse, using the same rendering options as for Figure 9.

measurement data of our tomography method on Grenoble and Toulouse was sufficient for the clustering algorithm to identify two clusters with 100% accuracy within the first 2 iterations (Figure 6), and this is in agreement with the used visualization (Figure 11).

D. Three and Four Site Experiments

In the following experiments, we use only intra-site nodes which are not separated by bottlenecks within their site (e.g. in the case of Bordeaux, all nodes used are in the well connected Borderline and Bordereau physical clusters).

First, we perform a three-site experiment, using the sites Grenoble, Bordeaux and Toulouse (32 nodes per site). Again, we perform 30 iterations, but only 2 iterations are sufficient for perfect accuracy (Fig. 6) of the modularity clustering. Three clusters are identified, which are also apparent in the visualization (Fig. 12).

In the experiment which spans most sites, we use 16 nodes for each of the sites Grenoble, Bordeaux, Toulouse and Lyon. Again, we perform 30 iterations. Modularity clustering of our BitTorrent tomography measurements correctly identifies the 4 logical clusters, which are also apparent in the visualization (Fig. 13). One interesting observation is that in this visualization, the central cluster of nodes represents the Lyon site, which is also positioned centrally in the star-like geographical topology of Figure 7. Also interesting is that in this four-

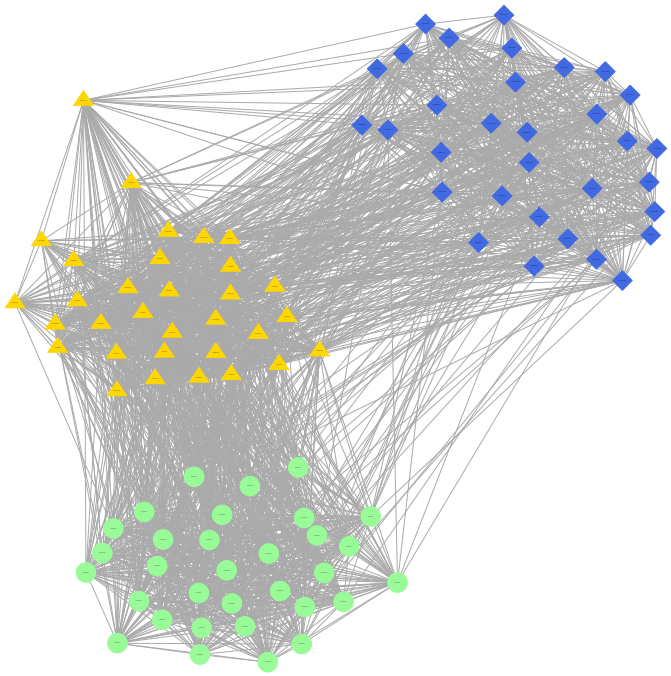

Fig. 12. Applying Kamada-Kawai layout to dataset 'BGT', with nodes in Bordeaux, Grenoble and Toulouse, using the same rendering options as for Figure 9.

site experiment we need around 15 iterations (See Fig. 6) to achieve perfect accuracy. While this is still very few, it is the largest number of iterations needed of all tested settings. This is not surprising as this is the setting with the largest number of logical clusters.

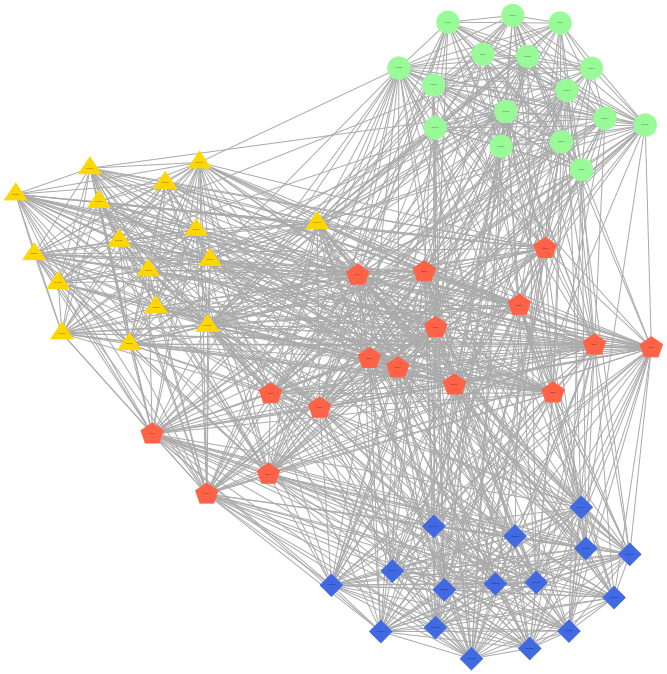

Fig. 13. Applying Kamada-Kawai layout to dataset 'BGTL', Bordeaux, Grenoble, Toulouse and Lyon, using the same rendering options as for Figure 9. The ground truth clusters in this rendering appear to be visually less distinctly laid out than the other examples; however, we note that the algorithmic clustering method still achieves perfect accuracy – see Figure 6.

V. Conclusion

In this work, we presented a novel approach to multiple source / multiple destination network tomography. Instead of using traditional bandwidth measurement techniques, we counted the number of fragments exchanged in BitTorrent broadcasts. Even a few iterations of this approach were sufficient to allow accurate reconstruction of the logical network clusters.

The reconstruction was done using a modern network clustering algorithm – modularity-based clustering. Our experimental results show that we can reliably find clusters with bandwidth tomography. Our approach is more efficient at revealing network properties which appear under intense collective communication than existing methods. Existing methods would take hours or days to uncover these details; our approach requires only a few minutes, and achieves high accuracy.

We also evaluated the number of BitTorrent broadcasts needed for various settings to achieve this accuracy. Correct clustering within a single site needed only a small number of iterations; whereas around 15 iterations are needed for our most complex experiment, taking only a few minutes, and running on a larger empirical network than studied in related bandwidth tomography literature.

Our approach correctly identified communication bottleneck links in physical clusters by placing the nodes communicating across the bottleneck link in different logical clusters - a significant result. It also separated nodes in different sites into different logical clusters due to the lower inter-site bandwidth.

The efficiency of our approach makes it useful for applications relying on bulk data transfer – e.g. applications performing all-to-all communication – across complex and heterogeneous networks of computers.

This automatic efficient measurement is also particularly suitable for overlay networks, or networks of virtual machines, which may have a dynamically altering underlying topology.

Our results were robust for all settings tested.

Future Work

We have seen promising results with this technique. A major advantage of the technique is that all parts of it – the BitTorrent-based measurement technique, and the clustering algorithm – are designed to scale to large networks. As we have shown that the clusterings found accurately uncover network structure in empirical networks for which ground truth is available, future work should integrate this tomography method into existing parallel computation libraries, and measure the performance increase gained on large networks. This would allow us further evaluate the effectiveness of this method in an application setting.

In this work, to keep our method compatible with existing software libraries, we have worked only with ground truths that are a single partitioning of the network into disjoint non-overlapping clusters. However, both the network clustering algorithm used, and the NMI evaluation method, extend to overlapping multi-level hierarchical clusterings. Extending our measurement technique and investigating the performance of the tomography approach on such hierarchical datasets would be valuable.

Acknowledgment

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant Number 08/IN.1/I2054 and 08/SRC/I1407.

Experiments presented in this paper were carried out using the Grid'5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr).

Special thanks to Sebastien Badia for clarifying the network topology within the Bordeaux site of Grid'5000.

References

[1] E. Gabriel, M. Resch, T. Beisel, and R. Keller, "Distributed computing in a heterogeneous computing environment," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, ser. Lecture Notes in Computer Science, V. Alexandrov and J. Dongarra, Eds. Springer Berlin / Heidelberg, 1998, vol. 1497, pp. 180–187, 10.1007/BFb0056574. [Online]. Available: http://dx.doi.org/10.1007/BFb0056574

[2] T. Kielmann, H. E. Bal, and S. Gorlatch, "Bandwidth-efficient collective communication for clustered wide area systems," in *Proc. 14th Int. Parallel and Distributed Processing Symp. IPDPS 2000*, 2000, pp. 492–499.

[3] N. T. Karonis, B. Toonen, and I. Foster, "MPICH-G2: A grid-enabled implementation of the Message Passing Interface," *Journal of Parallel and Distributed Computing*, vol. 63, no. 5, pp. 551 – 563, 2003, special Issue on Computational Grids. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731503000029

[4] C. Coti, T. Herault, and F. Cappello, "MPI applications on grids: A topology aware approach," in *Euro-Par 2009 Parallel Processing*, ser. Lecture Notes in Computer Science, H. Sips, D. Epema, and H.-X. Lin, Eds. Springer Berlin / Heidelberg, 2009, vol. 5704, pp. 466–477.

[5] H. Subramoni, K. Kandalla, J. Vienne, S. Sur, B. Barth, K. Tomko, R. Mclay, K. Schulz, and D. K. Panda, "Design and evaluation of network topology-/speed- aware broadcast algorithms for infiniband clusters," in *Proc. IEEE Int Cluster Computing (CLUSTER) Conf*, 2011, pp. 317–325.

[6] P. Bhat, C. Raghavendra, and V. Prasanna, "Efficient collective communication in distributed heterogeneous systems," in *Distributed Computing Systems, 1999. Proceedings. 19th IEEE International Conference on*, 1999, pp. 15 –24.

[7] J. Hatta and S. Shibusawa, "Scheduling algorithms for efficient gather operations in distributed heterogeneous systems," in *Proc. Int Parallel Processing Workshops*, 2000, pp. 173–180.

[8] R. L. Graham and G. Shipman, "MPI support for multi-core architectures: Optimized shared memory collectives," in *Proceedings of the 15th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 130–140. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-87475-1_21

[9] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: Recent developments," *Statistical Science*, vol. 19, no. 3, pp. pp. 499–517, 2004. [Online]. Available: http://www.jstor.org/stable/4144399

[10] Y. Tsang, M. Coates, and R. Nowak, "Passive network tomography using EM algorithms," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 3, 2001, pp. 1469 –1472 vol.3.

[11] V. N. Padmanabhan, L. Qiu, and H. J. Wang, "Passive network tomography using bayesian inference," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, ser. IMW '02. New York, NY, USA: ACM, 2002, pp. 93–94. [Online]. Available: http://doi.acm.org/10.1145/637201.637214

[12] R. Caceres, N. Duffield, J. Horowitz, D. Towsley, and T. Bu, "Multicast-based inference of network-internal characteristics: Accuracy of packet loss estimation," *IEEE Transactions on Information Theory*, vol. 45, pp. 2462–2480, 1998.

[13] M. Rabbat, R. Nowak, and M. Coates, "Multiple source, multiple destination network tomography," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 3, march 2004, pp. 1628 –1639 vol.3.

[14] L. Bobelin and T. Muntean, "Algorithms for network topology discovery using end-to-end measurements," in *Parallel and Distributed Computing, 2008. ISPDC '08. International Symposium on*, july 2008, pp. 267 –274.

[15] A. Legrand, F. Mazoit, and M. Quinson, "An application-level network mapper," LIP, Tech. Rep., 2002–09.

[16] H. Casanova, A. Legrand, and M. Quinson, "SimGrid: a generic framework for large-scale distributed experiments," in *10th IEEE International Conference on Computer Modeling and Simulation*, Mar. 2008.

[17] B. Cohen, "Incentives build robustness in BitTorrent," 2003.

[18] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, p. P10008, 2008.

[19] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[20] P. Rodriguez and E. W. Biersack, "Dynamic parallel access to replicated content in the Internet," *IEEE/ACM Trans. Netw.*, vol. 10, pp. 455–465, Aug. 2002. [Online]. Available: http://dx.doi.org/10.1109/TNET.2002.801413

[21] R. Xia and J. Muppala, "A survey of BitTorrent performance," *Communications Surveys Tutorials, IEEE*, vol. 12, no. 2, pp. 140 –158, quarter 2010.

[22] M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. Al Hamra, and L. Garcs-Erice, "Dissecting BitTorrent: Five months in a Torrent's lifetime," in *Passive and Active Network Measurement*, ser. Lecture Notes in Computer Science, C. Barakat and I. Pratt, Eds. Springer Berlin / Heidelberg, 2004, vol. 3015, pp. 1–11, 10.1007/978-3-540-24668-8_1. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24668-8_1

[23] Z. Liu and C. Chen, "A rule-based traffic exchange matrix estimation algorithm for BitTorrent tomography," in *Signal Processing, 2006 8th International Conference on*, vol. 4, 16-20 2006.

[24] S. Ren, E. Tan, T. Luo, S. Chen, L. Guo, and X. Zhang, "TopBT: A topology-aware and infrastructure-independent BitTorrent client," in *INFOCOM, 2010 Proceedings IEEE*, march 2010, pp. 1 –9.

[25] A. Zissimos, K. Doka, A. Chazapis, and N. Koziris, "Gridtorrent: Optimizing data transfers in the grid with collaborative sharing."

[26] Q. O. Snell, A. R. Mikler, and J. L. Gustafson, "Netpipe: A network protocol independent performance evaluator," in *in IASTED International Conference on Intelligent Information Management and Systems*, 1996.

[27] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information processing letters*, vol. 31, no. 1, pp. 7–15, 1989.

[28] J. Ellson, E. Gansner, E. Koutsofios, S. North, and G. Woodhull, "Graphviz and dynagraph static and dynamic graph drawing tools," *Graph Drawing Software*, pp. 127–148, 2004.

[29] A. Noack, "Modularity clustering is force-directed layout," *Physical Review E*, vol. 79, no. 2, p. 026102, 2009.

[30] M. Rosvall and C. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, p. 1118, 2008.

[31] B. Good, Y. de Montjoye, and A. Clauset, "Performance of modularity maximization in practical contexts," *Physical Review E*, vol. 81, no. 4, p. 046106, 2010.

[32] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, p. 033015, 2009.

[33] "Grid'5000 network wiki," Website, 2012, https://www.grid5000.fr/mediawiki/index.php/Grid5000:Network.